UKRAINIAN CATHOLIC UNIVERSITY

MASTER THESIS

# Fact editing in Large Language Models: in-weights vs in-context techniques

*Author:*
Oleksandr VASHCHUK

*Supervisor:*
Alberto BIETTI
Nazarii DRUSHCHAK

*A thesis submitted in fulfillment of the requirements for the degree of Master of Science*

*in the*

Department of Computer Sciences
Faculty of Applied Sciences


Lviv 2024

# Declaration of Authorship

I, Oleksandr VASHCHUK, declare that this thesis titled, "Fact editing in Large Language Models: in-weights vs in-context techniques" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

*"By far, the greatest danger of Artificial Intelligence is that people conclude too early that they understand it."*

Eliezer Yudkowsky

UKRAINIAN CATHOLIC UNIVERSITY

Faculty of Applied Sciences

Master of Science

**Fact editing in Large Language Models: in-weights vs in-context techniques**

by Oleksandr VASHCHUK

# *Abstract*

As Large Language Models (LLMs) have gained visibility for their ability to generate human-like text, ensuring the accuracy and reliability of the information they produce has become crucial. Thus facts-editing approaches received wide attention due to the possibility of editing the model's factual knowledge without investing resources to improve the dataset used for training the model, fine-tuning, or adaptive tuning. Together with the development of fact-editing methods also improves understanding of facts storage and retrieval inside the LLMs. The main goal of the work is to study factual retrieval mechanisms for in-context and in-weights knowledge.

This work concentrates on mechanistic interpretability for LLMs. Several experiments were conducted to understand the factual retrieval mechanisms in the model. As a result, important model components that contribute most during factual recall were identified.

# *Acknowledgements*

People and organizations in the following list contributed to the successfully completion of my master thesis. I want to thank and express great respect for their contribution:

- I want to thank my **advisor Alberto Bietti**. For constant support and help in the research. For cheering me up when I almost gave up. And for opening for me the great field of mechanistic interpretability. You truly inspire me.
- I want also thank my colleague student and **advisor Nazarii Drushchak**, for his bright ideas about structuring the research and notifications about upcoming deadlines. Thanks for your desire to help, appreciate it.
- Thanks to the **Faculty of Applied Sciences at UCU** for the great posibilites that team of the faculty provide. Thanks all the lecturers for your dedication in teaching. Also, I am grateful to **Ruslan Partsey** for his comments and help during the research.
- Special thanks to my beloved wife **Iryna**. It has been a difficult journey for both of us, but it would not have been possible without her support.
- Thanks to **Neel Nanda** and the team working on **TransformerLens**[1] library. This code reduced amount of needed time to setup an experiment.

I am very thankful to everyone who has helped me with their time, effort, and ideas. Your support was essential for finishing my thesis, and I truly appreciate it.

[1] https://github.com/neelnanda-io/TransformerLens

# Contents

# List of Figures

# List of Tables

# List of Abbreviations

| | |
|---|---|
| **LLM** | **L**arge **L**anguage **M**odel |
| **MLP** | **M**ulti **L**ayer **P**erceptron |
| **FFN** | **F**eed **F**orward **N**etwork |
| **AI** | **A**rtificial **I**nteligence |
| **RAG** | **R**etrieval **A**ugmented **G**eneration |
| **GPT** | **G**enerative **P**re-training **T**ransformer |
| **LLaMA** | **L**arge **La**nguage **M**odel Meta **AI** |
| **MHSA** | **M**ulti-**H**ead **S**elf **A**ttention |
| **ROME** | **R**ank-**O**ne **M**odel **E**diting |
| **MEMIT** | **M**ass-**E**diting **M**emory **I**n a **T**ransformer |
| **PMET** | **P**recise **M**odel **E**diting in a **T**ransformer |
| **IOI** | **I**ndirect **O**bject **I**dentification |
| **SOTA** | **S**tate **O**f **T**he **A**rt |

*Dedicated to every person who repels russian aggression.*
*Thank you. Slava Ukraini!*

# Chapter 1

# Introduction

Recent growth in popularity and usage of generative Artificial Intelligence (AI) models, especially Large Language Models (LLMs) — made this topic one of the most emerging ones in the field of Data Science. This significantly changed the way how people communicate, search for information and work with digital content. LLMs have an exceptional ability to generate texts very similar to human-written ones (Mann et al., 2020, Touvron et al., 2023, Jiang et al., 2023). Based on McKinsey research — generative AI features will add up to $4.4 trillion to the global economy annually in upcoming years [1].

The development of LLMs provided the ability to create sophisticated chatbots, and translations, generate and summarize content, it has also raised concerns regarding the reliability of the produced information. Text generated by LLMs is always questioned: Is it factually accurate? Can I trust it?

Since our world is changing constantly, there might be a need to edit, delete, or add certain facts to/in the model. Training such complex models requires a lot of time and computational power. The collection of datasets and their preparation also takes time and resources, and even more — training datasets are limited because existing data is finite (not all data can be used due to licensing issues). We need to have a method to edit the model without the costs associated with training models from scratch.

The concept of fact editing can help with cases when biases are present in training data, data is changed over time and new knowledge should be acquired, or information about fact is limited so it cannot be learned. It has a big impact on information integrity, public trust in digital content, and LLMs itself. If we as researchers and AI developers fail to address the rising concerns about the reliability of the produced information — it will decrease trust in online information sources, LLMs, and the digital ecosystem.

Text generation is often done with the usage of transformer-based LLMs. Transformer is a model architecture that avoids using recurrence and instead relies entirely on an attention mechanism to draw global dependencies between input and output (Vaswani et al., 2017). Attention mechanism which plays a key part in this architecture, is a complex method that gives the ability to focus on the most relevant parts of the input. The other key component of these models are feed-forward layers, which typically account for a large fraction of the model parameters, and are responsible for a large part of the knowledge storing (Geva et al., 2020).

In this thesis, we explore existing fact editing methods that modify fact in context. There exist several fact editing methods for in-model weights - ROME (Meng et al., 2022a), MEMIT (Meng et al., 2022b), PMET (Li et al., 2023). These methods

[1] https://www.mckinsey.com/featured-insights/mckinsey-explainers/whats-the-future-of-generative-ai-an-early-view-in-15-charts/

perform under the knowledge that inner MLPs act as knowledge storage (Geva et al., 2020).

Another common approach to improving factual recall is the so-called *retrieval-augmented generation (RAG)*, which augments the context of an LLM with documents containing relevant factual knowledge. We are thus also interested in how LLMs trade off factual knowledge that comes from their input context, versus from model weights (and thus training data).

In this work by examining the underlying mechanisms for factual storage and recall we aim to better understand existing circuits and their role. Currently, we know about several existing circuits (Wang et al., 2022) and attention heads roles (McDougall et al., 2023). Still, the gap exists because we don't clearly understand how the model is handling cases with in-context and in-model knowledge conflict (Xu et al., 2024).

Understanding the model behavior is crucial with such a rapid development of LLMs and a field. Currently, this instrument is used in a wide range of applications without a proper understanding of the tool. We suggest always checking the generated output for factual correctness. However, a better understanding of the model behavior will lead to increased trust in LLMs and generated output, which will help to improve existing fact-editing methods in models. Having effective methods will lead to ensuring ethical LLM development, where data which for example offend the honor and dignity of the individual could be edited precisely with reduced cost, as well as acquiring new knowledge.

Our motivation here is to know the tool that we are rapidly developing and adopting, and with this help future researchers who are working in the field of interpretability, because the ability to explain how and why specific output was generated is crucial to trust the tool

In this work, all experiments are performed on GPT-2 small. It is not the latest SOTA LLM but it still has general patterns that exist in other bigger models, and its size makes it easier to perform experiments and interpret the results. Still, this is considered as a limitation of the work, but it also gives us the ability to deliver comparable results to such works as a: Interpretability in the wild: a circuit for indirect object identification in GPT-2 small (Wang et al., 2022) and Locating and editing factual associations in GPT (Meng et al., 2022a)

To perform experiments the technique of activation patching is used. This technique was introduced in the ROME paper (Meng et al., 2022a) and was performed to determine the important layers that contain the biggest amount of knowledge that should be changed in the scope of intervention. We also adopted a technique called path patching from Interpretability in the wild paper (Wang et al., 2022) to locate attention heads circuits. All experiments are performed using TransformerLens[2] by Neel Nanda.

This work has the following structure: Chapter 2 gives information about background and related work. It describes the background related to mechanistic interpretability and fact-editing. Give an overview of related work in the domain. Chapter 3 describes research gaps, methodology, and techniques used in research. Chapter 4 is concentrated on data, existing datasets, and the process of new dataset creation. Chapter 5 outlines research results related to the interpretability of factual recall mechanisms. In chapter 6 discuss future work and limitations of the research.

[2] https://github.com/neelnanda-io/TransformerLens

# Chapter 2

# Background and Related work

## 2.1 Facts-related problems in Large Language Models

As the usage of LLMs spreads to different domains of human life the question arises — can we trust the LLM? In other words, is the generated output based on facts, which include both common knowledge and specific domain knowledge?

This "factuality" (Wang et al., 2023) problem is closely related to hallucinations (occur when LLM generates an output which has no sense, or is not related to prompt or is factually incorrect), outdated information, and domain specificity topics, which are broadly discussed right now.

To find a solution for these facts-related problems we need to understand how LLMs store factual knowledge to then use it for output generation.

Recent studies show that factual association is stored in the weights of early Multi-Layer Perceptrons (MLPs) (Meng et al., 2022a). Further research demonstrates that MLP value vectors store human-interpretable concepts and factual knowledge (Geva et al., 2023). It is also shown that the attention mechanism, to be specific Multi-Head Self Attention, works as a knowledge extractor and stores general knowledge extraction patterns (Li et al., 2023).

## 2.2 Knowledge conflict

Typical ways of using LLMs involve working with external contextual knowledge such as: user prompts, dialogues, and retrieved information from various sources like the Web or other tools. Using this contextual knowledge gives the ability for LLM to keep track of current events and generate more accurate responses, but it risks conflicting due to the rich knowledge sources. The discrepancies among the contexts and the model's parametric knowledge are referred to as knowledge conflicts (Xu et al., 2024).

There exist three types of knowledge conflicts:

- **Context-Memory conflict** arises when contextual knowledge conflicts with the knowledge stored in model weights;
- **Inter-Context conflict** happens as a conflict in different parts of the context. It can happen when the model uses different external sources of information. Some of them may not be secure, or contain outdated information, etc;
- **Intra-Memory conflict** is taking place due to the inconsistencies in the training data. Since questions could be described in various forms the responses from the model can change.

It is crucial to understand the mechanisms behind the knowledge conflicts to correctly suggest the approach to resolve them. Resolving these conflicts can improve the trustworthiness, accuracy, and robustness of LLMs (Xu et al., 2024). We will work on revealing the mechanisms behind the Context-Memory conflict.

There are two main causes of Context-Memory conflict:

- **Temporal misalignment** happens in cases when a model is presented with new information that happened after the time when training data was collected;
- **Misinformation pollution** arises in cases when context contains false or misleading information compared to what model was trained on;

For the **Misinfromation pollution** case, which is investigated in this work, the model deviates from the memory knowledge when presented with direct conflict in the context (Qian, Zhao, and Wu, 2023). Although, there are solutions that help resolve knowledge conflicts such as: fine-tuning, prompting, pre-training, and knowledge plug-ins; the interpretability side should be investigated.

## 2.3 Interpretability in Large Language Models

Mechanistic interpretability is used to discover, understand and verify the mechanisms that the model learns in its weights by reverse engineering model computations (Meng et al., 2022a, Geva et al., 2020). Understanding the internal model's mechanisms is crucial for understanding the model's behavior, it also helps to identify model errors and build effective algorithms to fix them.

Using mechanistic interpretability techniques recent studies show that factual associations are stored in the weights of early MLPs Geva et al., 2020. This research created a possibility to work on fact-editing algorithms such as Rank-One Model Editing (ROME) (Meng et al., 2022a) providing crucial knowledge about model mechanisms used for knowledge retrieval.

Some studies are working on attention heads role understanding. One of them discovered a mechanism named Copy suppression (McDougall et al., 2023) which reduces the model's ability to generate completion with the token from context. This copy-suppression mechanism consists of three steps:

- **Prior copying**. Components in the early layer predict that the completion should be done with the token that already appeared in context.
- **Attention**. Copy suppression attention heads detect this behavior and attend back to the previous instance of this token.
- **Suppression**. Copy suppression heads write directly to the model's output to decrease the generation probability of copied token.

Lowering the generation probability (logit) of the copied token, the steps above can increase the generation probability of correct completion. This work not only explains the role of attention head 7 on layer 10 (L10H7) but also states that the model tends to create completion by copying context from context. This aligns with the above-mentioned works which state that models are receptive to the information provided in context.

Another crucial work in the mechanistic interpretability domain is "Interpretability in the wild: a circuit for indirect object identification in GPT-2 small" (Wang et al.,

2022). This paper also provided a methodology framework for this work. Their main achievement is the discovery of a circuit (subset of the computation graph) which is responsible for completing the task of indirect object identification. Indirect Object Identification is a specific natural language task during which the model should predict the name that isn't the subject of the last clause. **Example:** *"When Mary and John went to the store, John gave a drink to"* should be completed with *"Mary"*.

By analyzing the circuit and attention head attention patterns authors discovered different groups of attention heads with specific tasks. Such tasks are:

- **Duplicate token heads** which identify tokens that already appeared in the sentence. These heads attend to the token previous occurrence and signal that token duplication happened;
- **S-inhibition heads** remove a duplicate token from Name Mover Heads' attention;
- **Name mover heads** output the remaining name (here names are tied to the specific task, on which authors worked on). These heads attend to the previous name and copy it to the output;

Another interesting finding is so-called Backup heads which perform the task when main heads are ablated. This shows the model's complex structure, as the circuit can change when some other component is ablated. There are also attention heads identified that write in the opposite direction of the correct answer, the assumption behind this is that these heads help model reduce loss during training.

These interpretability researches help to understand better components of the model and define specific tasks which component is performing. Further research in this field can help with the development of advanced algorithms for fact editing in LLMs and also model behavior modification. For example, increasing the importance of the Copy Suppression mechanism can decrease model susceptibility to information provided in the context.

## 2.4 Editing facts in Large Language Models

There are several reasons why models fail to generate output based on factual association: model-level, retrieval-level, and inference-level causes. This review will concentrate on model-level causes which are: domain knowledge deficit, outdated information, immemorization, forgetting, and reasoning failure (Wang et al., 2023).

The concept of model editing was proposed (De Cao, Aziz, and Titov, 2021) and stated as neural language memory edit that influences the output of the model when given the input which is related to edited fact. It is the change in the model's parameters that affects predictions for specific input. Besides that, model editing methodologies should meet the desired criteria such as:

- **Generality**: The method should apply to models which were not specifically designed to be editable;
- **Reliability**: The method should change a fact and not affect model knowledge;
- **Consistency**: Changes should be effective to similar ways of stating the fact (Yao et al., n.d., De Cao, Aziz, and Titov, 2021);

There are several popular approaches to model editing nowadays that can be split into two main streams.

**Methods that keep LLM's parameters:**

- **Memory-based model approach:** This approach stores edit examples directly in memory and adds an extractor to get the most relevant fact edit for the specific input. It gives the possibility to generate output with edited facts or return unchanged output if it is not relevant to any fact edit stored in memory. There are also several methods based on in-context learning. Since LLMs can retrieve knowledge from the provided context — there are methods built on top of this capability. These methods change the output of the model by adding edited facts to the prompt (Yao et al., n.d.).
- **Additional parameters approach:** This approach extends the LLM's inner state by adding additional parameters. Several methods patch the last Feed-Forward Network layer of the model with one or a couple of neurons per fact change or mistake correction (Yao et al., n.d.).

**Methods that change LLM's parameters:**

- **Meta-Learning:** This set of methods is based on the key idea of creating a hyper-network that learns the needed difference in weights that should be applied to the language model (Yao et al., n.d.).
- **Locate-Then-Edit:** This approach first works on identifying parameters that are in charge of storing specific facts. After that, modification is applied to the LLM's parameters. There are several methods to achieve modification. For example, Knowledge Neurons (Dai et al., 2021) introduces a method of locating key-value pairs in the Feed-Forward Network and then updating these key-value pairs values to alter the model's knowledge. Another set of methods started from ROME (Meng et al., 2022a) locate layers that store factual associations. Then the MLP weights matrix is changed to alter the stored knowledge.

To build methods that change LLM's parameters researchers first have to understand the underlying mechanisms for knowledge retrieval. That is why mechanistic interpretability is so important, it helps to build methods and improve their effectiveness.

## 2.5 Locate-then-edit methods detailed overview

In this section, we want to overview the most recent and powerful methods in the locate-then-edit family, which achieved the best results so far. These methods are developed using the latest advancements in the field of mechanistic interpretability. This review shows the current efficiency of the methods, further research in mechanistic interpretability should help to achieve better results.

ROME method (Meng et al., 2022a) besides the editing also introduces the causal tracing method. Causal tracing is used to identify the layers that contribute more than others to retrieve the facts about input text. This method is based on causal mediation analysis (Meng et al., 2022a). To measure each state contribution to a fact retrieval, in this method all model's hidden states activations are observed during three runs:

1. Starting run which predicts the output on a given input, this run is used to collect all hidden states activation values.

2. This run is using noised input. The subject is corrupted by adding noise with normal distribution. Thus, the model loses data stored in the subject tokens and produces a set of hidden states activations. Generated output with noised input will be incorrect.

3. The last run checks the ability of a single state to restore the model's ability to generate correct output using noised input. At this step — the model's hidden states activations from the second run are substituted with activation values collected during the first run at some specific token and layer.

| Editor | Score | Efficacy | Generalization | Specificity | Fluency | Consistency |
|---|---|---|---|---|---|---|
| GPT-J (6B) | 22.4 | 15.2 (0.7) | 17.7 (0.6) | 83.5 (0.5) | 622.4 (0.3) | 29.4 (0.2) |
| ROME | 50.3 | 50.2 (1.0) | 50.4 (0.8) | 50.2 (0.6) | 589.6 (0.5) | 3.3 (0.0) |
| MEMIT | 85.8 | 98.9 (0.2) | 88.6 (0.5) | **73.7** (0.5) | 619.9 (0.3) | 40.1 (0.2) |
| PMET | **86.2** | **99.5** (0.1) | **92.8** (0.4) | 71.4 (0.5) | **620.0** (0.3) | **40.6** (0.2) |
| GPT-NeoX (20B) | 23.7 | 16.8 (1.9) | 18.3 (1.7) | 81.6 (1.3) | 620.4 (0.6) | 29.3 (0.5) |
| MEMIT | 82.0 | 97.2 (0.8) | 82.2 (1.6) | **70.8** (1.4) | **606.4** (1.0) | 36.9 (0.6) |
| PMET | **84.3** | **98.4** (0.2) | **89.4** (0.5) | 70.3 (0.5) | 598.1 (0.6) | **38.9** (0.2) |

TABLE 2.1: 10,000 counterfact edits on GPT-J (6B) and GPT-NeoX (20B). Within parentheses is the 95% confidence interval.(Li et al., 2023)

Thus the ability of some specific layer to restore the model's generated output to correct one will indicate its causal importance. To measure the indirect effect for a specific state, the difference between the probability of the generation of correct output at run 3 and the probability of the generation of correct output at run 2 is calculated.

ROME aims to make rank one update to the MLP part of the model. First of all, by using the causal tracing method the most important MLP layer is located. Since the above-mentioned work stated that MLP is acting as a key-value store, the ROME method is solving the constrained least-squares problem to insert an updated key-value pair into MLP's hidden state.

Mass Memory Editing in a Transformer (MEMIT) (Meng et al., 2022b) is a continuation of the ROME method with several alterations that make it more suitable to edit more than 1 fact per edit. Using the causal tracing method the range of important MLPs is selected. That means that desired factual knowledge change could be shared between those important MLP layers. To perform edit — the target MLP layer at the end of important layers is chosen. At this target layer, the edited factual knowledge should be fully represented. Then, for each important layer, the modification is performed to apply approximately equal portion of an edit.

Precise Model Editing in a Transformer (PMET) (Li et al., 2023) is a continuation of the MEMIT method. To determine where the update should happen, PMET uses the same technique as MEMIT. PMET simultaneously modifies hidden states of MLP and MHSA in the target layer and modifies hidden states of MLP for other important layers. PMET follows the same steps of editing as in MEMIT, but has a key difference — it updates the target layer MLP and MHSA. Also, PMET doesn't distribute the updates in approximately equal portions, it uses square root spread to add more updates to important layers. They show that Multi-Head Self Attention layers work as continuous extractors of different types of knowledge, while MLPs extract only factual knowledge encoded in their weights matrix. They also conclude that adding a new fact doesn't need an MHSA layer update (Li et al., 2023).

The latest results of these methods application you can review in Table 2.1 introduced in the PMET paper.

As you can see in Table 2.1 efficiency scores for these methods are high, and they are capable of editing facts in model weights.

## 2.6 Conclusion

Mechanistic interpretability plays a key role in the understanding models' behavior and helps to improve trustworthiness and robustness. It also gives the ability to develop improved algorithms and mitigation strategies for different problems. As was discussed in this chapter - mechanistic interpretability research created a push toward fact-editing methods in LLMs.

# Chapter 3

# Methodology

In this chapter, used methodology, research objectives, and research flow are presented. Our main focus is to study the mechanisms at play for factual recall from both weights and context, as well as the potential competition between the two.

## 3.1 Significant tokens logits visualisation

To better understand how tokens are promoted on different layers of the model in this work specific visualisation was used. The output of each layer was normalized and multiplied by an unembedding matrix. This gives the ability to understand how the probability of generation for a specific token is changed over the layers of the model.

In Figure 3.1 we show the example of this visualization usage. It can show how on which layer the propagation of the specific token differs from others, which can show an important layer, which is crucial for token generation. To understand which component in that layer is causing such behavior - activation patching is used.

## 3.2 Activation patching

To locate the model's components which are responsible for some specific action the technique of activation patching is used. This technique is done to identify the model's specific hidden states that have the biggest impact on the generated tokens. These states form a computational graph that contains many paths from inputs to the output (generated token).

To determine each state's individual contribution we observe internal activations during three runs: run on the original prompt, run on a new prompt that could be an original prompt corrupted by noise or a new prompt with preserved structure but changed context, and a run with restoration that gives the ability to calculate an influence of a specific state to restore the expected generation result.

- **Original prompt run** we pass the prompt that causes the model to perform the task that we want to investigate. During this run, all needed hidden activations (by that we mean MLP layer outputs, attention layer outputs, attention head query, key our value based on the investigated component) are cached.

- **New prompt run** is performed on a new prompt that differs from the original prompt. In this case, the original prompt could have obfuscated information by adding generated noise to the set of tokens (Meng et al., 2022a), or have a structure of the original prompt but with broken relations between entities in it (Wang et al., 2022). The model runs on this new prompt resulting in a set of changed activations. Since in this run model loses the information from the

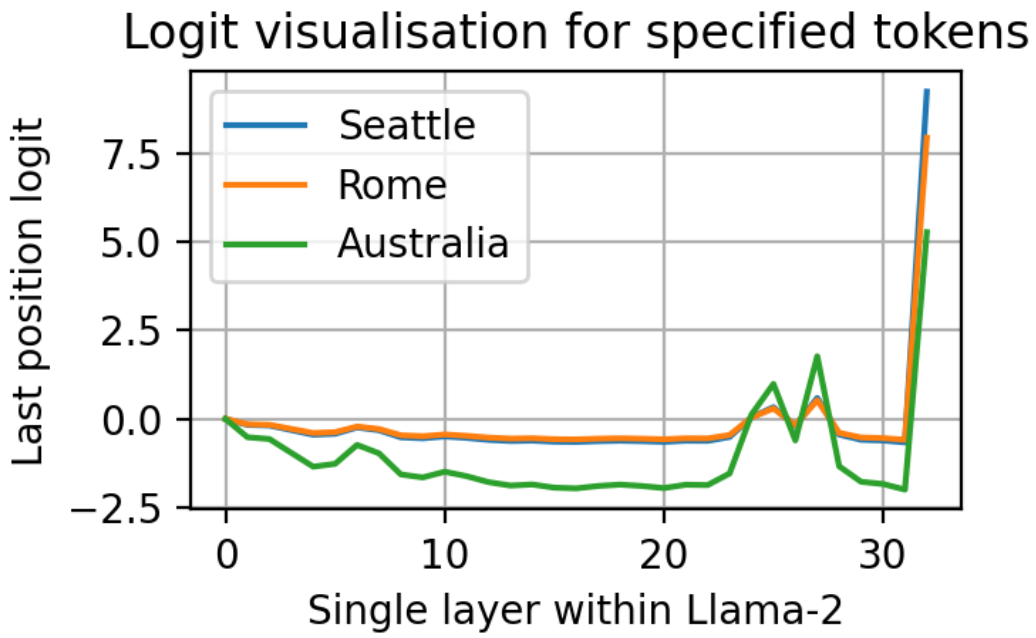



FIGURE 3.1: Example of logits visualization on the last token position. Three specific tokens are tacked in this example. Value on the y axis is the logit value before normalization.

original prompt it will result in the generation of a token that differs from the token generated in the first run.

- **Run with restoration** is performed on a new prompt, except that at some token and layer component activations are substituted by those calculated during run on the original prompt. That allows us to determine the ability of a component to recover the expected token, when other states are performing with activations computated based on a new prompt. This ability will indicate the component's importance in the computation graph.

The importance of the component is calculated based on a difference between logit values for the expected token from run with restorations and run on a new prompt. This difference shows the impact on the generation results from the tested component.

## 3.3 Definition of circuits and knockouts

In our work, we rely on the concept of circuits defined in Interpretablity in the wild paper (Wang et al., 2022) as a useful abstraction that gives the ability to understand the correspondence between the components of a model and human-understandable concepts. Representing the model as a computation graph when nodes are terms in its forward pass and edges are the interactions between those terms, a circuit is a subgraph of a computational graph responsible for some behavior (Wang et al., 2022).

To identify circuits knockouts are used. A knockout removes a set of nodes in a computation graph to eliminate the node's impact on the generation but capture all other computations in a model. The circuit is defined by knocking out all nodes in

the computation graph and taking the resulting logit outputs in the modified computational graph (Wang et al., 2022).

In our work, we are using mean ablation for the node knockout. It means that the activation value of the node is replaced by the average activation value across the reference distribution. Mean-ablation method removes the information that varies in the reference distribution but preserve constant information (Wang et al., 2022).

Method of **mean ablation** was chosen in favor of **zero ablation** for our work because it shows more robust results. Zero ablation means that node activation is set to 0. Since 0 is an arbitrary value, and following nodes can rely on the average activation value as an implicit bias term. Zero ablation leads to noisy results in practice (Wang et al., 2022).

## 3.4 Path patching

To trace back the information flow for the identified important attention head we performed the path patching technique suggested in the same Interpretability in the wild paper (Wang et al., 2022). This technique is used to identify which attention heads indirectly influence affect model's logits through defined attention head.

In this technique we observe internal activations for attention heads during four runs: the first run collects the activations and computes average values for the ablation on the new prompt; the second run collects activations on the original prompt; the third run knockout with mean ablation the selected attention heads preserving the outputs of the target attention head on the original prompt; fourth run performs the generation with restoration of the activations collected on the third run on target head on original prompt.

- **First run**. Performs a run on the reference distribution which preserves the structure of the prompt but eliminates the information from it. During that run, the activations are preserved (queries, keys, or values) and the average value of the activation is computed.
- **Second run**. Performs a run on the original prompt, all activations are preserved.
- **Third run**. Performs a run on the original prompt, during which attention heads — predecessors for the target head are mean ablated one by one. Other predecessor heads have frozen values from the second run. Target head activations are collected since they reflect the output with intervention into the circuit.
- **Fourth run**. Performs a run on the original prompt, during which target attention head activations are restored from the collected values on the fourth run.

Using this technique we can identify attention heads that indirectly influence the output of the model and thus form a specific circuit.

## 3.5 Research gaps. Discovery of factual recall mechanisms.

This work focuses on identifying various mechanisms for factual recall with or without context-memory conflict. Compared to ROME, we would like to improve our

understanding of attention layers for knowledge extraction, in particular, the way attention layers combine several parts of the input sequence to retrieve a fact, in particular, the following:

- *relations*, e.g., through a verb such as "located in" or "owned by";
- *in-context* facts, e.g., when facts appear in a document included in the context, as in RAG applications;
- *modifiers*, which might change the behavior of the model, e.g., including the text "ignore the previous context" to encourage the model to forget about the context and use its weights instead.

To achieve this, we will combine causal tracing techniques from the ROME paper, together with circuit identification techniques (e.g., from the IOI circuit paper (Wang et al., 2022)). We are planning to use various existing libraries to help us with these goals, in particular the ROME repository[1], and the TransformerLens[2] library by Neel Nanda.

## 3.6 Research flow

As this work concentrates on cases with context-memory conflict, first of all, we need to create a dataset that will allow us to investigate this case. Existing datasets are very tied to the specific research domain and are hardly scalable to another domain, for example, it's hard to use CounterFact (discussed in detail in Chapter 4) in the mechanistic interpretability research for specific task. Based on this information first step of the research is to create a dataset that will fulfill the needs.

As a second step, the series of experiments will be conducted with logits visualization for a set of tokens (altered knowledge item provided in the context and factual knowledge that should be retrieved) and activation patching for layers output and attention heads. This step will allow to understand the patterns in the model's output generation by checking the logits visualizations (we are specifically interested in understanding on which layer one token is more promoted or suppressed rather than another). Then using activation patching on layers and specific attention heads will show important (those which gives the biggest impact on the generation result) layers and components (attention heads, attention layers, MLP layers).

The third step is the analysis of the results. On these steps we will be working on understanding how these important layers and components are working together on the specific task. Review attention patterns for identified attention heads, apply path patching to identify circuits that are responsible for retrieving altered knowledge under context-memory conflict conditions or retrieving fact knowledge from model weights.

As a deliverables of this work should be:

- created dataset for investigation of the specific task;
- code implementation of techniques mentioned above, as well as code needed for experiment setup;
- highlights on the circuits that are used in the process of token generation under context-memory conflict;

[1] https://github.com/kmeng01/rome

[2] https://github.com/neelnanda-io/TransformerLens

- highlights on the circuit that is used for the generation of token which is not in the context, the specific interest in this case is next: are attention heads used during the generation of this token?

## 3.7 Conclusion

This chapter gives an overview of the methodology which will be used in this work. The main tools and frameworks which will be used, are described here. This methodology will address research objectives and provide rigorous analysis for information retrieval under context-memory conflict.

A methodology was assembled based on a review of the latest advancements in mechanistic interpretability topic. Also, datasets were reviewed, and the demand for creating a separate dataset was identified.

In summary, the methodology chapter provides a detailed overview of the research approach for this study.

# Chapter 4

# Data

## 4.1 Existing datasets and metrics

To make this research comparable, existing datasets and metrics used for the evaluation of results were reviewed. The most common datasets used for the fact editing task are: zsRE (Levy et al., 2017) and CounterFact (Meng et al., 2022a). The Indirect Object Identification (IOI) dataset was also reviewed, there are no links in the original paper (Wang et al., 2022) to this dataset, but there is a dataset on HuggingFace [1] which was inspired by IOI research (Brian Muhia, 2022).

zsRE dataset consists of the fact statement, paraphrase of this statement, and neighborhood statement (Mitchell et al., 2021). zsRE dataset is used to assess editing methods' ability to correct facts (De Cao, Aziz, and Titov, 2021). The main aim of editing methods is to change the specific factual knowledge and correctly generate output for paraphrased facts without changing the knowledge for neighborhood statements. To measure the effectiveness of this change for this dataset the next metrics are used:

- Efficacy — the share of the cases that generate changed factual knowledge.
- Paraphrase — the share of the cases that generate changed factual knowledge for a paraphrased fact.
- Specificity — the share of cases that generate correct factual knowledge for neighborhood facts.
- Score — harmonic mean for Efficacy, Paraphrase and Specificity (Meng et al., 2022b).

The CounterFact dataset consists of a fact rewrite entry, two paraphrased entries, a set of neighborhood entries, and prompts that implicitly rely on fact. This dataset was designed for better generalization tracking for fact editing methods. To measure the effectiveness of editing methods the next metrics are used:

- **Efficacy score (ES)** the portion of cases for which we have a probability of emitting edited factual knowledge over previous state knowledge.
- **Efficacy magnitude (EM)** is the mean difference between the probability of emitting edited factual knowledge and previous state knowledge.
- **Paraphrased score (PS)** and **Paraphrased magnitude (PM)** computed similarly to ES and EM but for the paraphrased input equivalent to the one which was used for editing.

[1] https://huggingface.co/

- **Neighbourhood Score (NS)** and **Neighbourhood Magnitude (NM)** computed similarly to ES and EM but for the nearby subject prompt which should result in the generation of the edited knowledge.
- **Score (S)** harmonic mean of ES, PS, NS.
- **Consistency (RS)** as the cos similarity between the unigram TF-IDF vectors of generated texts, compared to reference texts about subjects sharing edited factual knowledge.
- **Fluency (GE)** weighted average of bi- and tri-gram entropies (Meng et al., 2022a).

The IOI dataset consists of two patterns for prompts: BABA and ABBA, where A and B are names of the Subject1 and Subject2. Each of these patterns has a set of templates that describe different situations with a specific object.

**Example:**

- **BABA**: "Then, B and A went to the PLACE. B gave a OBJECT to" A should be predicted.
- **ABBA**: "Then, A and B went to the PLACE. B gave a OBJECT to" A should be predicted.

As a metric, for this dataset is used logits difference for two tokens. In this specific task, these tokens are named A and B. To be used for ablation purposes - this dataset also has an ABC transformation. That means that the template is preserved, but names are no longer connected, example: "Then, A and B went to the PLACE. C gave a OBJECT to". This allows to preserve general information/patterns and remove data that varies.

These datasets are used in recent works and were investigated to understand the common approach in dataset creation and results evaluation. This work is concentrated on context-memory conflict cases and the existing dataset doesn't cover these cases. This is the motivation for us to create additional dataset for this field.

## 4.2 Dataset

Since this work concentrates on the specific information retrieval task it was decided to create a dedicated dataset to work with. Before this decision, existing datasets were reviewed and analyzed.

Datasets as zsRE or CounterFact (Meng et al., 2022a) are specifically designed for the fact editing task, but this dataset gives the essential knowledge about the desired structure of the dataset. This dataset doesn't consist of the prompt itself, but it also contains a set of metadata such as: the subject of the prompt, expected generation output, expected generation output after editing, neighborhood prompts, and generation prompts (for results evaluation).

In comparison, the dataset for Indirect Object Identification (Brian Muhia, 2022) is built for a specific task investigation. It isn't shared publicly, but there is a dataset that is generated from instructions written in a paper. It contains much less metadata.

For this research, a new dataset specific to this task should be generated. During the exploratory analysis part of the research it was decided to work on the case with context-memory conflict. In this case, we define a conflict in the first part of

the prompt, and then the prompt should be completed with a token from context. **Example:** "Alanya is in the country of Uganda. The city of Alanya is located in country of". In this example, the first part (we also call it in-context learning) of the prompt is altering the location of the mentioned city to another country. The second part of the prompt is the completion part where we expect the model to generate country mentioned country. The geography theme was selected due to the good model knowledge about geography.

For this dataset, we have the following templates:

- "A is in the country of B. The city of A is located in country of" - original prompt with which experimentation started.
- "The city of A is located in the country of B. The city of A is located in the country of" - prompt with an improved copying pattern to reveal the copy mechanism in the model.

This dataset consists of two parts - the original one with the structure mentioned above and, the second so-called ABC. ABC stands for that cities and countries are not aligned at all. **Example:** "A is in the country of B. The city of C is located in country of". By averaging component values over these entries, we generate values for the knockout (mean ablation).

This dataset also contains metadata. Each entry in the dataset have: fact output value (country in which city is located); expected output data (country mentioned in the first part of the prompt); cities mentioned in the prompt; and countries mentioned in the prompt. Besides the metadata and prompt mentioned above, each entry has a paraphrase:

- "A is in the country of B. The city of A is located in country of" - where B is the correct country for the mentioned city A. This prompt has context and memory information aligned.
- "A is in the country of B. The city of C is located in country of" - where B is the correct country for the mentioned city A, and C is another city from another country. This prompt has context and memory information aligned, it also breaks relations between two parts of the prompt.

Prompt paraphrases were used in different experiments during the research. Since they preserve the structure of the prompt, these items could be used for different ablations to understand specific model mechanisms.

To perform experiments we generated 100 of samples for each part of the dataset. Each city is mentioned in the whole dataset only once. Sample consists of 16-19 tokens based on the city or country used (some cities or countries can have more than 1 word).

In details, you can review the dataset on GitHub [2].

## 4.3 Data processing

Data processing for the dataset could be divided into different revisions that will be discussed below:

[2] https://github.com/vashchuko/context-memory-conflict

- **First revision.** The main complexity was to find a list of cities and countries to correctly structure dataset entries. At first cities from Opendatasoft [3] were taken. This list contains cities with a population of more than 1000. To build dataset entries, cities were randomly selected from the list, and prompts with the first template were created.

  The problem with this revision is that for a subset of prompts - the model didn't generate correct completion. After investigation, it was found that some of the cities taken to the dataset, even don't have their own wiki page. Under the assumption that the model could not have enough training data to be aware of the city, it was decided to create a new revision of the dataset.

- **Second revision.** A new list of cities was found from Simplemaps [4]. This time, entries in the dataset were generated with the first template and satisfied the next conditions:

  - Cities should have more than 250k population (to ensure its wide presence in the training data).
  - Each prompt and its variations were tested on the target model to verify that the generated token is the expected one.

- **Third revision.** To identify the model's copying circuit it was decided to align two parts of the prompt and regenerate the dataset with a second prompt template under the conditions mentioned above.

## 4.4 Conclusion

In this section, the dataset creation was discussed. This dataset is generated to fulfill the need to research model circuits for a specific task. It is accessible on GitHub to ensure repeatability of the research.

[3] https://public.opendatasoft.com/explore/embed/dataset/geonames-all-cities-with-a-population-1000/table/?disjunctive.cou_name_en&sort=name

[4] https://simplemaps.com/data/world-cities

# Chapter 5

# Experiments

The main objective of this chapter is to present experiments done to evaluate models of factual retrieval mechanisms under context-memory conflict. These experiments aim to address outlined research questions. Through these experiments, we seek to uncover the inner model's mechanisms that will contribute to a deeper understanding of model behavior.

## 5.1 Model Selection

At first, this project was targeted to LLaMa2 model. But then it was switched to a smaller one — GPT-2 small. There were several reasons for this transition.

- Existing works (Wang et al., 2022, McDougall et al., 2023) in the field of mechanistic interpretability are concentrated on smaller models. To make this work comparable it was decided to work with GPT-2 small.
- LLaMa2 configuration starts from 32 layers and 32 attention heads at each layer [1]. It results in 1024 attention head patterns that should be reviewed and analyzed. This significantly slows the research process. On the other hand —- GPT2-small have 12 layers and 12 heads [2]. This results in 144 attention heads, which need less time for the review and analysis.
- Since the research in the field of mechanistic interpretability for the LLMs is at the beginning of its development — it is crucial to understand key patterns of model behavior. Then experiments could be scaled to bigger models to understand how these patterns are incorporated there. Given that bigger models still share the same architecture, we have the assumption that previously identified attention heads with specific tasks and circuits are implemented in bigger models as well. However, scaling features remain unknown.

GPT-2 small usage is considered a limitation for this research, but it also offers an advantage to reuse existing libraries like TransformerLens [3] as well as previously discovered attention head tasks and circuits.

[1] https://huggingface.co/docs/transformers/en/model_doc/llama2
[2] https://huggingface.co/docs/transformers/en/model_doc/gpt2
[3] https://github.com/TransformerLensOrg/TransformerLens

## 5.2 Exploration phase

This research started from the exploration of the ROME fact editing method (Meng et al., 2022a) and its scaling to LLaMa2. The main interest was to understand techniques, such as Activation patching, used to determine important MLP components that should be changed by the ROME method.

We performed a set of activation patching experiments on the CounterFact dataset using LLaMa2 [4] and GPT2-XL[5] models. During these experiments — the subject of the sentence was corrupted by the random noise, and components were patched with activations retrieved from the uncorrupted run.

On the Figure 5.1 you can see that, the important MLP layers in LlaMa2 are also located at the beginning of the path and are related to the subject tokens. However, Multi-Head Self Attention (MHSA) acts a bit differently than in GPT-2-XL. Here GPT-2-XL was selected to make these figures more comparable in terms of model size. This figure also proves the assumption that, despite the model size, architecture remains the same and mechanisms used to retrieve knowledge are similar.

Figure 5.2 shows logit visualization on each layer on the last token position for the set of selected tokens for various examples. Analyzing results we saw interesting patterns: fluctuation in logits for expected token over the layer; and a jump of logits value at the last layer. At that point in time we couldn't explain such a model's behavior, but wanted to investigate which model components contribute to this activity.

ROME method is editing only one MLP component (Meng et al., 2022a), we consider this as a limitation, because it doesn't take into account other components like Attention heads. There is a further development of ROME in the PMET method (Li et al., 2023) where Attention component weights are modified as well as MLP component, but it is done without understanding each head task or contribution. Despite that — PMET method achieves better results than ROME, as it is shown in Table 2.1.

This led us to another set of experiments where we constructed a prompt with an in-context learning part (altering the city's geoposition) to understand more about the involvement of the Attention component.

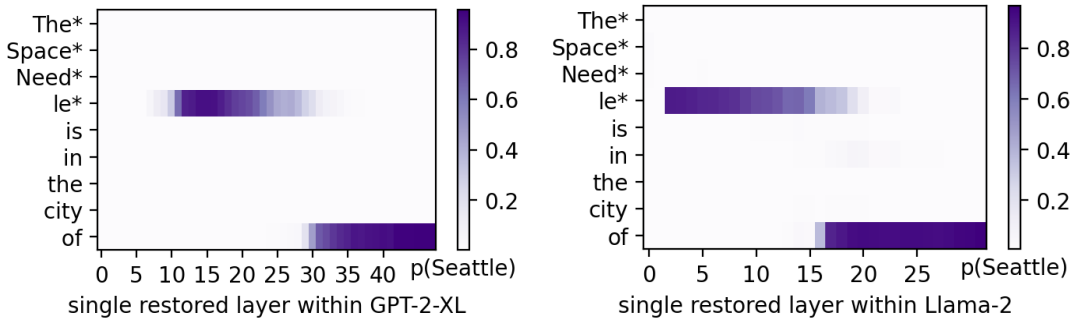
## 5.3 Narrowing the research to context-memory conflict

During this phase with several iterations, we constructed the prompt *"Rome is in the country of Barbieland. The city Rome is located in the country of"*. This prompt has context-memory conflict. In the first part the real country of Rome is changed to fake Barbieland, second part aims to retrieve the changed location. The model generated Barbieland as the location for Rome as expected. Logits visualization is in Figure 5.3.

This behavior is expected since models often deviate from the memory knowledge when presented with direct conflict in the context (Qian, Zhao, and Wu, 2023). But since we know that factual association is stored in the weights of early MLPs (Meng et al., 2022a) then the assumption is that the Attention component is responsible for copying behavior. To reveal this behavior we conducted a set of experiments

[4] https://colab.research.google.com/drive/15pKGLLZo_wWbk-pis8IMKgaAyycsWfYU?usp=sharing

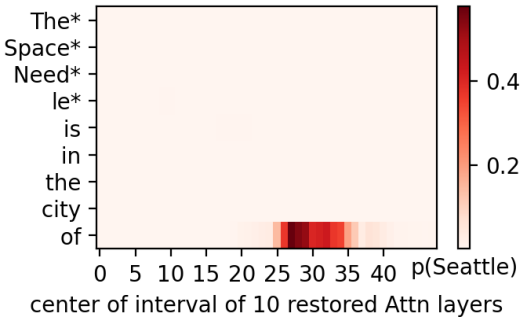
[5] https://colab.research.google.com/drive/1yLFygpYXTVzSN2egRouVClZueCJm1YJP?usp=sharing



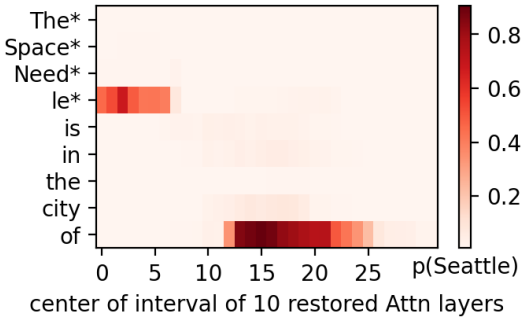

(a) Impact of restoring state after corrupted input



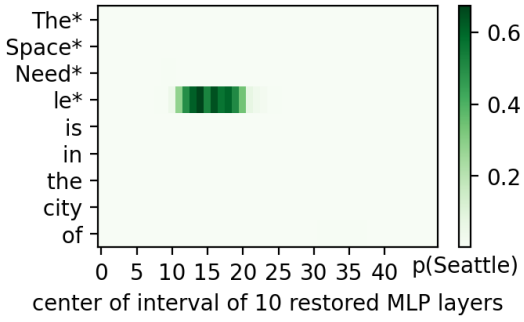

(b) Impact of restoring state after corrupted input



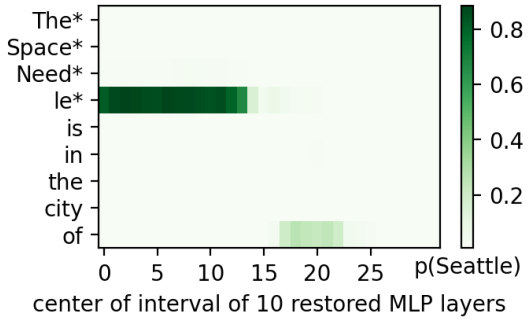

(c) Impact of restoring Attn after corrupted input




(d) Impact of restoring Attn after corrupted input




(e) Impact of restoring MLP after corrupted input



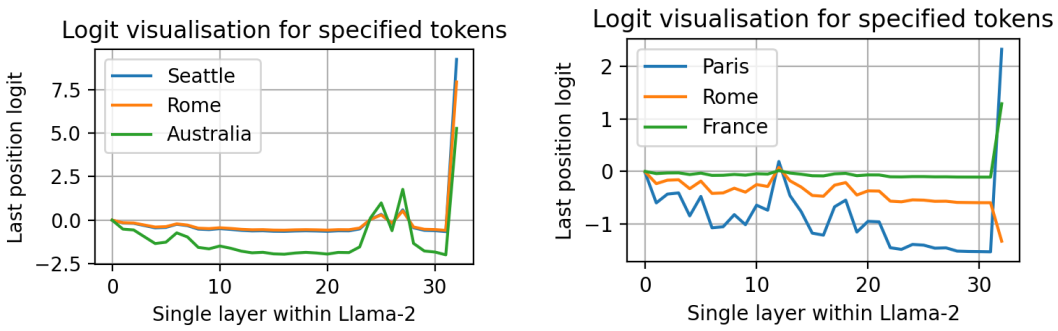

(f) Impact of restoring MLP after corrupted input

FIGURE 5.1: Activation patching comparison between GPT-2-XL (1st column) and LlaMa2 (2nd column). Figures show the indirect effect on output probability mapped for the contribution of (a, b) each hidden state on the prediction, (c, d) only attention activations, (e, f) only MLP activations.

using activation patching for this prompt, to understand which components show the biggest indirect effect on the generation.

Two experiments [6] were done: 1. First appearance of the Rome token is noised. With this setup, we want to test if this token is important for building a "relation" with the next token appearance. 2. The second appearance of the Rome token is noised. With this setup, we want to test if the "relation" will be created. Besides specific assumptions that we had about Rome tokens, the main aim of this experiment was to understand which components are important for the generation.

[6] https://colab.research.google.com/drive/15pKGLLZo_wWbk-pis8IMKgaAyycsWfYU?usp=sharing




(a) Prompt: "The Space Needle is in the city of"

(b) Prompt: "Vichy France's capital,"

FIGURE 5.2: Logit visualization on the last token position. Three specific tokens are tacked.

Logit visualization on Figure 5.4 shows that adding noise to the first Rome appearance indeed breaks the relation and the model generates the correct Rome location. Surprisingly, adding noise to the second Rome appearance didn't break the relation, and the model generated updated knowledge. We can explain this behavior using the concept of *binding IDs* (Feng and Steinhardt, 2024. We assume that even though the second Rome appearance was corrupted by the noise, the model built a binding ID using the first part of the sentence and generated the expected token. Although this is an assumption, proving it due to the time constraint is postponed for future work.

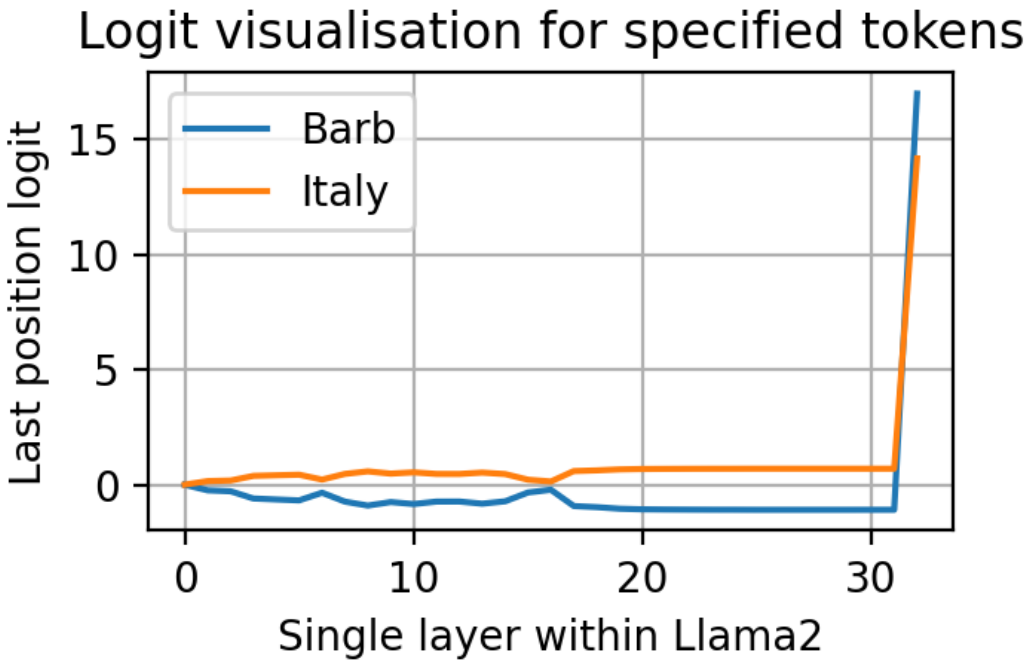
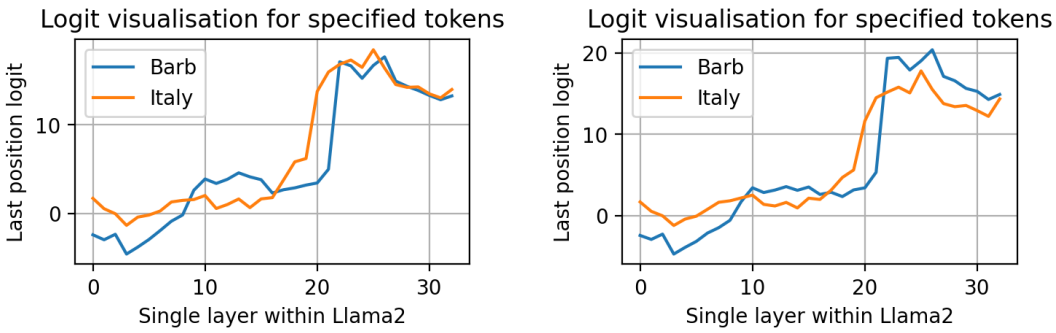



FIGURE 5.3: Logits visualization on the last token position. Prompt: "Rome is in the country of Barbieland. The city Rome is located in the country of". Barbieland has bigger logits in the result as expected




(a) First Rome appearance have added noise (b) Second "Rome" appearance have added noise

FIGURE 5.4: Logit visualization on the last token position.

Analyzing the component's indirect effect on generation results we draw the following conclusions:

- For both experiments we can see a clear pattern for the attention component. Attention components are more important on a second Rome appearance position, rather than on a last token position. The assumption here is that on the second Rome position — attention heads are working on retrieving binding IDs from the first part of the prompt. On the last token position, attention heads are important because the model accumulated information, and a decision on what should be generated is made here. We assume that it is a place where context-memory conflict is happening and the attention component is responsible for resolution.
- Important MLP components differ for both experiments. We can see that late layers MLP components are important on the last token position. While early layers MLP components are important on first and second Rome appearance. The purpose of early layers MLP components is storing factual knowledge (Geva et al., 2020), late layers MLP components, we assume, work on narrowing down the model output to the expected token in multidimensional space of embeddings.

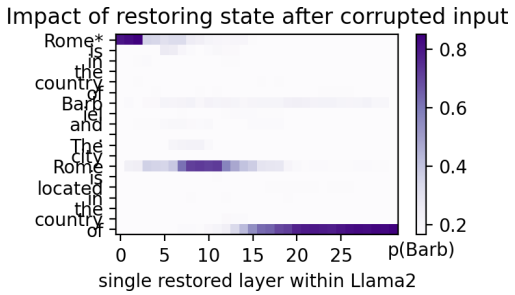
To investigate in-depth attention components we need to conduct experiments on the head level and scale them to the bigger amount of prompts.

## 5.4 Results

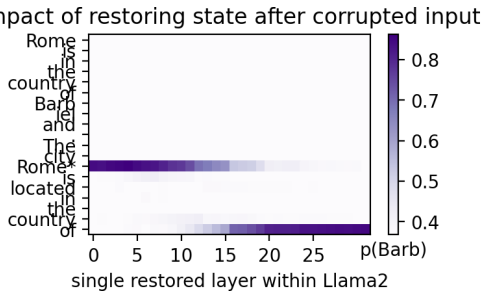
There were several attempts to conduct experiments on the created dataset, as well as reconstruct path patching methodology from the IOI paper (Wang et al., 2022). These experiments are done using GPT2-small, since working with LLaMa2 was very extensive by time and computation resources, as well as the amount of data that should be analyzed was times bigger due to the model size. In this section, we will provide the latest results and the main obstacles that we faced during the path.

**Main obstacles:**

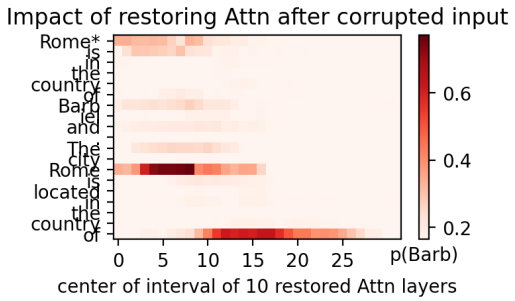
- After the first set of activation patching experiments we found attention heads that contributed most to the expected token generation. To investigate in depth



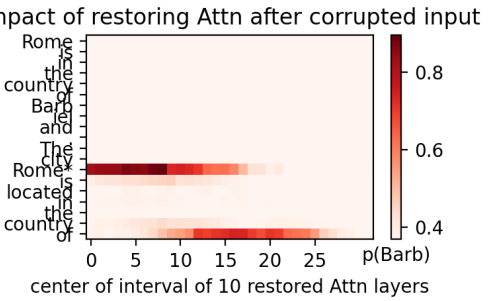

(a) Impact of restoring state after corrupted 1st Rome appearance in input



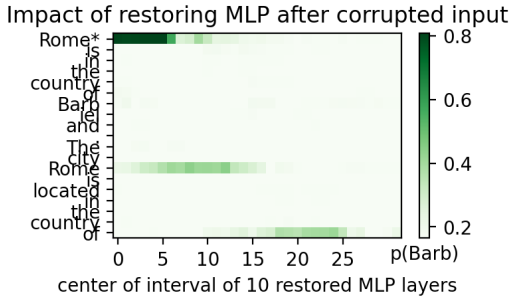

(b) Impact of restoring state after corrupted 2nd Rome appearance in input



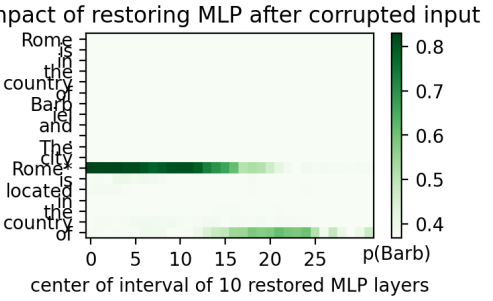

(c) Impact of restoring Attn after corrupted 1st Rome appearance in input




(d) Impact of restoring Attn after corrupted 2nd Rome appearance in input




(e) Impact of restoring MLP after corrupted 1st Rome appearance in input




(f) Impact of restoring MLP after corrupted 2nd Rome appearance in input

FIGURE 5.5: Activation patching done on LLaMa2 comparison between inputs with noised first (left column) and second (right column) Rome appearance in the prompt. Figures show the indirect effect on output probability mapped for the contribution of (a, b) each hidden state on the prediction, (c, d) only attention activations, (e, f) only MLP activations.

the circuit behind them — we aimed to reproduce the path patching methodology used in the IOI paper. Despite a separate appendix in the corresponding paper specifically dedicated to this technique, it was unclear which component values (attention query, attention value, attention output, mlp output) this technique was used for which experiment. It was also unclear which activation values should be frozen (restored) on each step. These parameters were reconstructed by us during experimentation. We hope that the next researchers who will be using this technique will benefit from our description in

the Methodology chapter.

- It is crucial to ensure that the model can generate the expected token from the given prompt. In our case, the first revision of the dataset had cities with very low populations, and as a result very low probability of being present in the training data. The results on this revision of the dataset were skewed and confusing, so we believe that the absence of such constraint is a mistake that leads to a waste of time and computation resources.

The latest results will be shared for the second and third revisions of the dataset. The first revision showed skewed results. Applying the path patching technique to the found important attention heads didn't show any significant circuits. Failure of this technique was the first sign that the setup of the experiment was incorrect and led to building a set of constraints that should be met during dataset generation.

Detailed discussion on dataset revision was done in corresponding Chapter 4, but here it is crucial to remind that:

- 2nd dataset revision has verified prompts that generate an expected token. It also has more known cities (based on the population size). **Example:** *"Bristol is in the country of Liberia. The city of Bristol is located in country of"*;
- 3d dataset revision has the same criteria applied to prompts verification and city selection. It has changed the prompt template to promote the copying behavior of the model. **Example:** *"The city of Ottawa is located in the country of Yemen. The city of Ottawa is located in the country of"*;
- ABC distribution is the part of the dataset, where each prompt is constructed using a relevant prompt template, but cities and countries are not related to each other. Averaging component activations on this distribution gives the value for mean ablation;

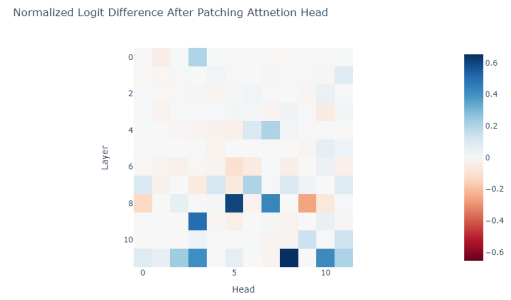
There were two experiments set up to reveal mechanisms that are working during context-memory conflict:

- 1. As the input of the model, the original prompt was used. Then specific component was ablated using mean ablation with value derived by averaging activations of this component during runs on ABC distribution[7]. This experiment was designed to analyze components that take active part in token generation under context-memory conflict;
- 2. As the input of the model, the paraphrase of the original prompt is used. Cities in the prompt are no longer connected, the first part of the prompt contains the correct country for the mentioned city. Then specific component was ablated in the same way as in the first experiment[8]. This experiment was designed to analyze components that take active part in token generation from model knowledge and suppression of dummy copying behavior (when model is copying the mentioned country without tying it to the mentioned city;

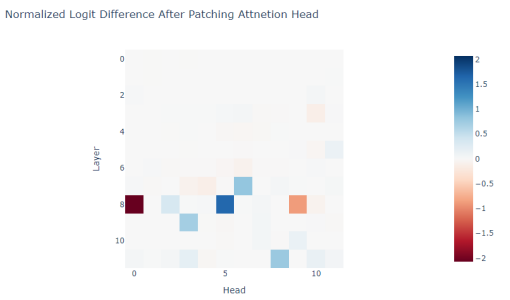
For the mentioned experiments we applied ablation to attention head values (we worked also on query values, but it shows results only for specific heads like name mover heads (Wang et al., 2022) and it is left for future work), MLP and Attention layers output values.

[7] https://colab.research.google.com/drive/1XsKfhoy1EX22VARLBKgkTmywdM1hZ70F?usp=sharing

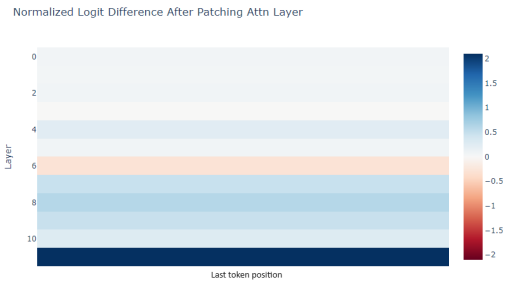
[8] https://colab.research.google.com/drive/1IHctp6huEG5sdpIQAlgv8p8dk1HzdNh-?usp=sharing



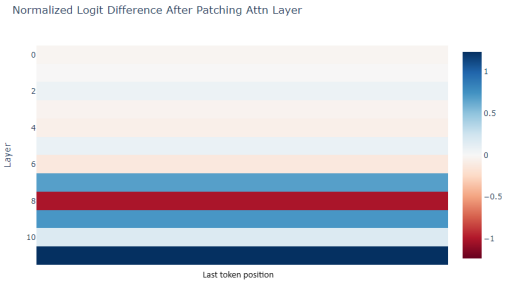

(a) Impact of mean ablation on the attention head level, dataset 2nd revision



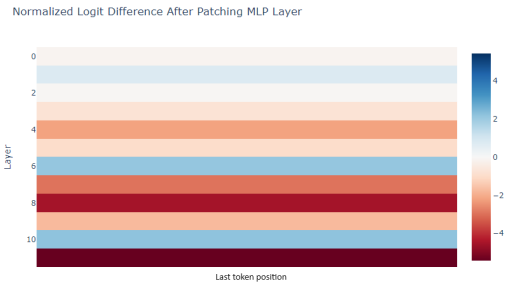

(b) Impact of mean ablation on the attention head level, dataset 3d revision



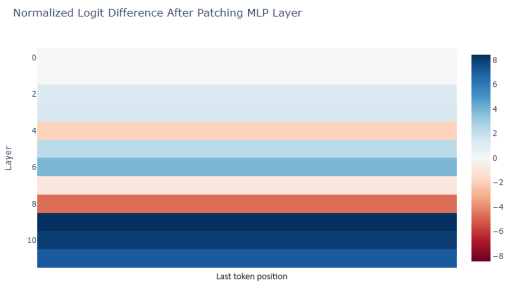

(c) Impact of mean ablation on the attention layer, dataset 2nd revision




(d) Impact of mean ablation on the attention layer, dataset 3d revision

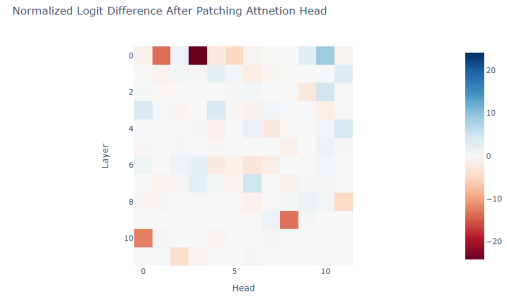



(e) Impact of mean ablation on the MLP layer, dataset 2nd revision

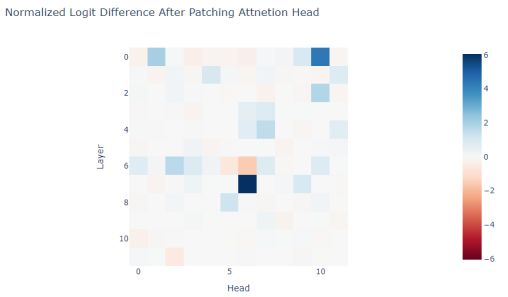



(f) Impact of mean ablation on the MLP layer, dataset 3d revision

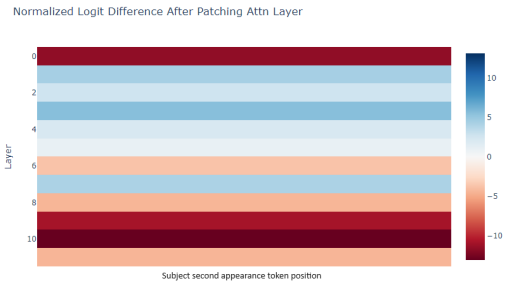
FIGURE 5.6: **1st experiment.** Mean ablation done on GPT-2-small, comparison between 2nd dataset revision (left column) and 3d dataset revision (right column) on the **last token position**. Figures show the indirect effect on output probability mapped for the contribution of (a, b) attention head, (c, d) only attention activations, (e, f) only MLP activations.



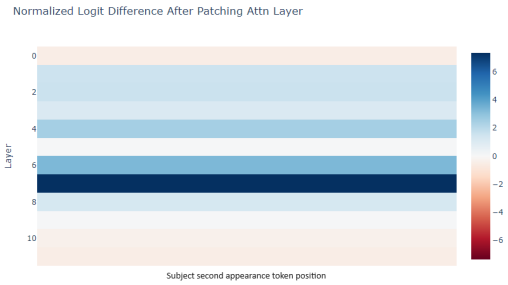

(a) Impact of mean ablation on the attention head level, dataset 2nd revision



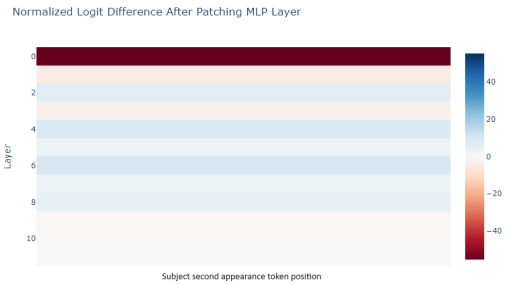

(b) Impact of mean ablation on the attention head level, dataset 3d revision



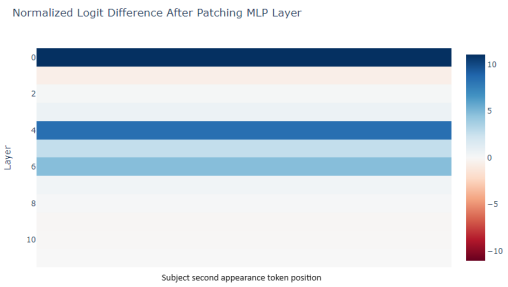

(c) Impact of mean ablation on the attention layer, dataset 2nd revision




(d) Impact of mean ablation on the attention layer, dataset 3d revision

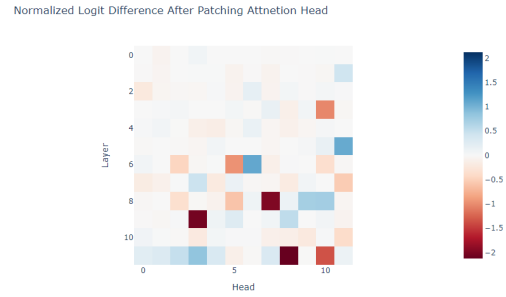



(e) Impact of mean ablation on the MLP layer, dataset 2nd revision

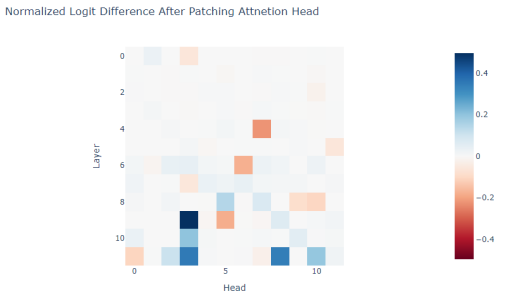



(f) Impact of mean ablation on the MLP layer, dataset 3d revision

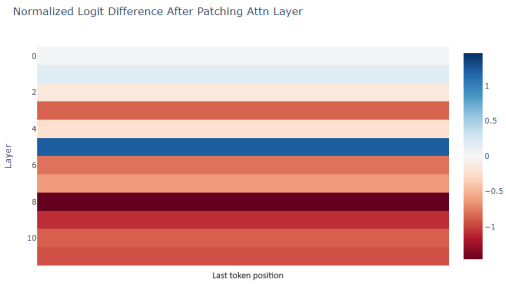
FIGURE 5.7: **1st experiment.** Mean ablation done on GPT-2-small, comparison between 2nd dataset revision (left column) and 3d dataset revision (right column) on the **subject second appearance token position**. Figures show the indirect effect on output probability mapped for the contribution of (a, b) attention head, (c, d) only attention activations, (e, f) only MLP activations.



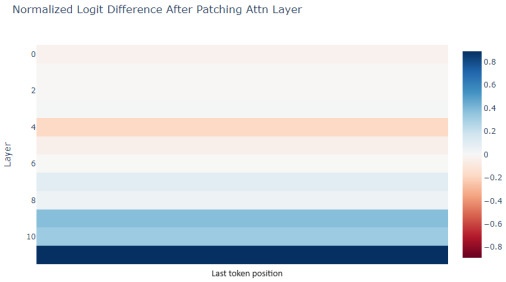

(a) Impact of mean ablation on the attention head level, dataset 2nd revision



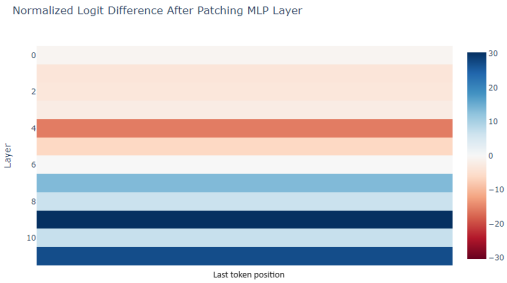

(b) Impact of mean ablation on the attention head level, dataset 3d revision



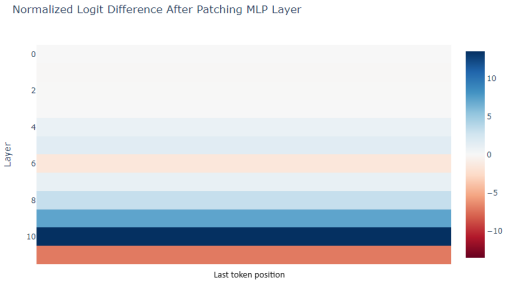

(c) Impact of mean ablation on the attention layer, dataset 2nd revision




(d) Impact of mean ablation on the attention layer, dataset 3d revision

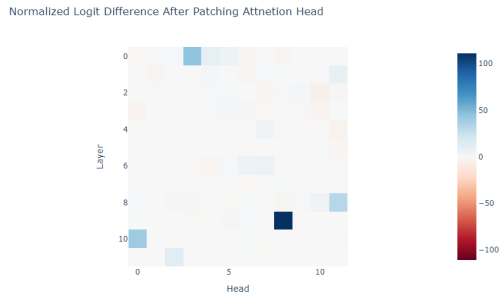



(e) Impact of mean ablation on the MLP layer, dataset 2nd revision

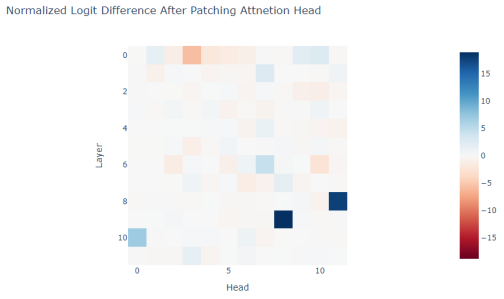



(f) Impact of mean ablation on the MLP layer, dataset 3d revision

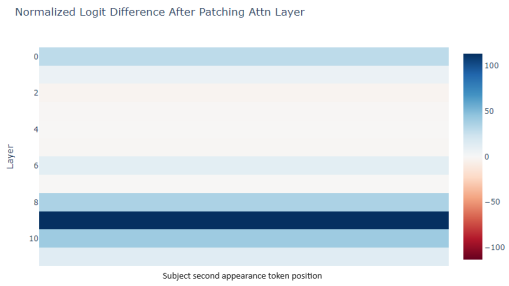
FIGURE 5.8: **2nd experiment.** Mean ablation done on GPT-2-small, comparison between 2nd dataset revision (left column) and 3d dataset revision (right column) on the **last token position**. Figures show the indirect effect on output probability mapped for the contribution of (a, b) attention head, (c, d) only attention activations, (e, f) only MLP activations.



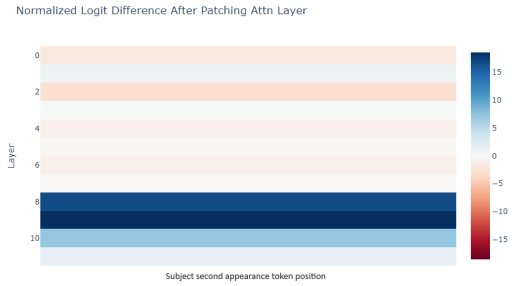

(a) Impact of mean ablation on the attention head level, dataset 2nd revision



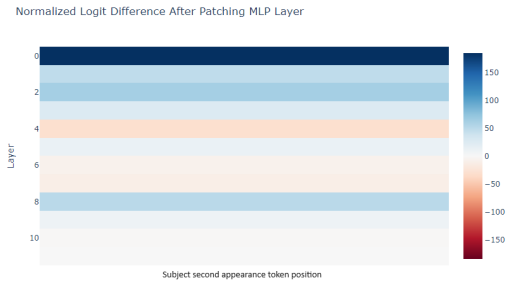

(b) Impact of mean ablation on the attention head level, dataset 3d revision



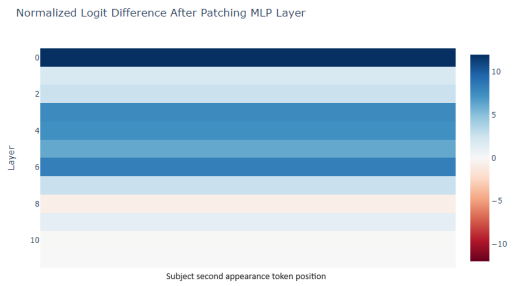

(c) Impact of mean ablation on the attention layer, dataset 2nd revision




(d) Impact of mean ablation on the attention layer, dataset 3d revision




(e) Impact of mean ablation on the MLP layer, dataset 2nd revision




(f) Impact of mean ablation on the MLP layer, dataset 3d revision

FIGURE 5.9: **2nd experiment.** Mean ablation done on GPT-2-small, comparison between 2nd dataset revision (left column) and 3d dataset revision (right column) on the **subject second appearance token position**. Figures show the sum indirect effect on output probability mapped for the contribution of (a, b) attention head, (c, d) only attention activations, (e, f) only MLP activations.

Results of these experiments you can see on Figure 5.6, Figure 5.7, Figure 5.8 and Figure 5.9. The value presented in the figures is the sum of the indirect effect on token generation probability after ablation. The indirect effect on token generation probability is calculated by the formula: *1 - ALD/OLD*. Where: *Ablated Logit Difference (ALD)* is — first token logit value minus second token logit value on a run with ablation; *Original Logit Difference (OLD)* is — first token logit value minus second token logit value on a run without ablation. For the first experiment — *first token* is a country mentioned in context (expected output), *second token* is the correct country where the city is located. For the second experiment — *first token* is a correct country where the city is located (expected output), *second token* is a country mentioned in context. In the presented figures we are interested in blue-colored components because ablation applied to these components decreases logit difference, which means that these components worked as first token promoters or second tokens suppressors.

Due to time limitations, we didn't apply path patching to the important heads discovered during experiments to reveal existing circuits in the model. As well we didn't apply unembedding to the attention head values to reveal what is written to the residual stream. These two tasks are left for future work.

### 5.4.1 First experiment results

Our main conclusions from the results for **first experiment** (context-memory conflict investigation):

- Results shows that the sum of indirect effect on token generation probability is significantly higher on the subject second appearance token position (second mention of the city in prompt, Figure 5.7) in comparison to a last token position (Figure 5.6). We saw that activity during previous experiments with activation patching done differently. That gives us solid confidence that this is the most important place in the computation graph.
- We also see that the sum of the indirect effect on token generation probability is significantly higher for MLP layers rather than Attention Layers. It explains why ROME (Meng et al., 2022a) and MEMIT (Meng et al., 2022b) achieve so great results in fact editing. And also is aligned with insights derived by a team that constructed the PMET method (Li et al., 2023). In their work, they derived the next conclusion: *Multi-Head Self Attention (MHSA) weights store certain general knowledge extraction patterns along with a small amount of factual knowledge* (Li et al., 2023).
- Analyzing indirect effect of MLP layers on **subject second appearance token position** we see that the first MLP layer contributes differently in dataset revisions (Figure 5.7 subfigures **e**, **f**). Recent research[9] says that early MLPs recognize entities and produce their attributes as directions with help of attention heads on the first levels. We can see that attention heads 1 and 3 on layer 0 (L0H1, L0H3) have a big negative effect aligned with the corresponding MLP layer (Figure 5.7 subfigures **a**, **b**). Investigation of this different contribution is left for future work.
- Revision 3 of the dataset gives much more understandable results and simpler attention head map rather than revision 2 (Figure 5.7 subfigures **a**, **b**) so our work will be continued with this template as it isolates copying circuit better.

[9] https://www.alignmentforum.org/posts/iGuwZTHWb6DFY3sKB/fact-finding-attempting-to-reverse-engineer-factua

- Investigation of important attention heads attention patterns and output is left for future work. But we are specifically interested in the task of: L8H0, L8H5, L11H8 on the **last token position** (Figure 5.6 subfigure **a**, **b**); L0H1, L0H3, L0H10, L7H6, L9H8 on **subject second appearance token position** (Figure 5.7, subfigure **a**, **b**)

### 5.4.2 Second experiment results

Our main conclusions from the results for **second experiment** (factual retrieval investigation):

- In this case we also see that sum of indirect effect on token generation probability is significantly higher on **subject second appearance token position** (Figure 5.9) in comparison to a last token position (Figure 5.8). However, it is not technically the second mention of the **same** city in a prompt.
- As in the first experiment we see that the sum of the indirect effect on token generation probability is significantly higher for MLP layers rather than Attention Layers, but only **on the last token position**. This is not true for the **subject second appearance token position**, we don't have an explanation of this phenomenon right now, it is left for future work.
- In comparison to the first experiment analysis of the indirect effect of MLP layers on **subject second appearance token position** shows that the first MLP layer contributes equally in both dataset revisions (Figure 5.9 subfigures **e**, **f**). Since there is no context-memory conflict in this experiment we assume that prompt structure here doesn't play a key role.
- As in first experiment, revision 3 of dataset give much more understandable results, less extreme values (Figure 5.9 subfigures **c**, **e**) and simpler attention head map rather than revision 2 (Figure 5.8 subfigures **a**, **b**).
- Investigation of important attention heads attention patterns and output is left for future work. Here we are specifically interested in the task of: L9H3, L11H8, L11H10 on the **last token position** since they show opposite behavior on different dataset revisions (Figure 5.8 subfigure **a**, **b**); L8H11, L9H8 on **subject second appearance token position** since they show the biggest effect (Figure 5.9, subfigure **a**, **b**)

The conducted experiment's results align with other papers in the field but also disclose a very important fact — specific circuit investigation is very tied to the constructed prompt templates. We expect to reveal circuits that are responsible for context-memory conflict resolution after working with attention heads attention patterns and validating their output in future research.

## 5.5 Conclusions

During this research, we investigated the area of fact editing methods and mechanistic interpretability fundamentals. After that, based on the recent work in the mechanistic interpretability area we reproduced several experiments (activation patching in ROME). Created dataset and worked on circuits investigation for context-memory conflict by conducting several experiments. Although there is a lot of future work to

be done to answer all the questions raised from our results, still we have these key deliverables:

- Created dataset for our task research;
- Code which implements core methodologies like activation patching, path patching, and logit visualization and is shared in collab notebooks;
- Series of experiments whose results align with current knowledge in the field and identified important components and token positions for future research;

# Chapter 6

# Conclusion

## 6.1 Discussion and Future Work

As the result of our work we achieved to locate the important attention heads for both our experiments — context-memory conflict and factual recall. We also deliver dataset and offer additional check for it to work with geographical representations under this setup. With this investigation we now understand at which token model forms inner representation which then is specified to return factual information.

As future work we will continue conducted experiments and analysis of its results.

1. Research defined attention heads attention patterns, to understand what is the task of the specific head, how it contributes to the final token generation;
2. Research defined attention heads output value, to understand what these heads generate, how this output is then distributed to other components (next attention heads and MLP components in the computation graph);
3. Run path patching technique on identified attention heads to understand which other components (mainly attention heads) indirectly contribute through identified attention heads. That will allow isolate the circuit and test it using minimality and completeness criteria;
4. Run experimets on bigger GPT-2 models to understand how these models differ in terms of important components;
5. Present refined results as a separate paper to the wider audience;

There are other ideas for future work closely related to fact-editing algorithms like:

- *Weight editing*, apply ROME/MEMIT/PMET and investigate changes in activated circuits. It will help better understand how changes in specific set of components imply model circuits and research compensation mechanisms;
- *Fine-tuning*, instead of manually editing a model's weights, we will fine-tune the model (or potentially a subset of its layers) on a dataset of sequences containing the new facts. Here we are also interested in understanding how fine-tuning different layers may affect the factual recall circuits;
- *Prompting*, where the instruction token which force model to ignore previous info is included in the LLM context (with various prompting strategies to be compared), and may thus modify the model's prediction.

Such a study will help us better understand how different mechanisms affect factual recall behaviors, as well as the contribution of different internal layers. But all those ideas worth dedicated research based on their complexity.

## 6.2 Ethical considerations

Our work is concentrated on the improvement of our understanding of LLMs factual recall mechanisms. Revealing these mechanisms should improve trustworthiness and understanding of model behavior. But this knowledge can reveal potential mechanisms that will allow modify LLMs in a harmful way, that can lead to unexpected results. We advise to always use models from well known vendors.

Also, all editing methods have the potential to be harmful by using them to add incorrect facts. We strongly advise that LLM's output should be carefully reviewed to be sure that it is factually correct.

## 6.3 Limitations

This work focus on GPT-2 models, specifically GPT-2-small, which have smaller amount of parameters than current state-of-the-art models. Interpretability of experiments and derived conclusions are dependent on embedding space of the model's inner components. This approach is widely used in mechanistic interpretability research (Wang et al., 2022, McDougall et al., 2023). Due to a time constraint not all our assumptions are proven by responding experiments, we aim to do that in following research.

# Bibliography


Brian Muhia (2022). *ioi (Revision 223da8b)*. DOI: 10.57967/hf/0142. URL: https://huggingface.co/datasets/fahamu/ioi.

Dai, Damai et al. (2021). "Knowledge neurons in pretrained transformers". In: *arXiv preprint arXiv:2104.08696*.

De Cao, Nicola, Wilker Aziz, and Ivan Titov (2021). "Editing factual knowledge in language models". In: *arXiv preprint arXiv:2104.08164*.

Feng, Jiahai and Jacob Steinhardt (2024). *How do Language Models Bind Entities in Context?* arXiv: 2310.17191 [cs.LG].

Geva, Mor et al. (2020). "Transformer feed-forward layers are key-value memories". In: *arXiv preprint arXiv:2012.14913*.

Geva, Mor et al. (2023). "Dissecting recall of factual associations in auto-regressive language models". In: *arXiv preprint arXiv:2304.14767*.

Jiang, Albert Q et al. (2023). "Mistral 7B". In: *arXiv preprint arXiv:2310.06825*.

Levy, Omer et al. (2017). "Zero-shot relation extraction via reading comprehension". In: *arXiv preprint arXiv:1706.04115*.

Li, Xiaopeng et al. (2023). "Pmet: Precise model editing in a transformer". In: *arXiv preprint arXiv:2308.08742*.

Mann, Ben et al. (2020). "Language models are few-shot learners". In: *arXiv preprint arXiv:2005.14165*.

McDougall, Callum et al. (2023). "Copy Suppression: Comprehensively Understanding an Attention Head". In: arXiv: 2310.04625 [cs.LG].

Meng, Kevin et al. (2022a). "Locating and editing factual associations in GPT". In: *Advances in Neural Information Processing Systems* 35, pp. 17359–17372.

Meng, Kevin et al. (2022b). "Mass-editing memory in a transformer". In: *arXiv preprint arXiv:2210.07229*.

Mitchell, Eric et al. (2021). "Fast model editing at scale". In: *arXiv preprint arXiv: 2110.11309*.

Qian, Cheng, Xinran Zhao, and Sherry Tongshuang Wu (2023). *"Merge Conflicts!" Exploring the Impacts of External Distractors to Parametric Knowledge Graphs*. arXiv: 2309.08594 [cs.CL].

Touvron, Hugo et al. (2023). "Llama 2: Open foundation and fine-tuned chat models". In: *arXiv preprint arXiv:2307.09288*.

Vaswani, Ashish et al. (2017). "Attention is all you need". In: *Advances in neural information processing systems* 30.

Wang, Cunxiang et al. (2023). "Survey on factuality in large language models: Knowledge, retrieval and domain-specificity". In: *arXiv preprint arXiv:2310.07521*.

Wang, Kevin et al. (2022). "Interpretability in the wild: a circuit for indirect object identification in gpt-2 small". In: *arXiv preprint arXiv: 2211.00593*.

Xu, Rongwu et al. (2024). "Knowledge Conflicts for LLMs: A Survey". In: arXiv: 2403.08319 [cs.CL].

Yao, Yunzhi et al. (n.d.). "Editing large language models: Problems, methods, and opportunities. CoRR, abs/2305.13172, 2023b. doi: 10.48550". In: *arXiv preprint arXiv.2305.13172* ().